

# Procedural LOD

# Procedural LOD

— [ Downside to procedural approaches: dealing with the impact on the game?

— [ Especially when used to amplify user input

— [ With a non-procedural game, apply the "stick of clue" to level designer, artist, or engineer who brings the game engine to its knees.

— [ Not an option for our games

# The Sims: Lots

- [ User can build extremely elaborate lots
- [ Can also decorate lots with fences etc.
- [ Deform terrain, paint with brushes
- [ Provide a neighbourhood view of many lots without bringing the game to its knees?
- [ In Sims 1: an artist-generated rendering from Maya

# Examples



# Examples



# Examples



# Examples



# Sims 2 Neighbourhood

- [ Collapse detailed wall, floor, and roof geometry into simple polys (Use game data structures)
- [ Use render-to-texture to imposterize walls and more
- [ Windows get baked into walls – not explicitly represented
- [ Same with fences, railings, stairs, even pools
- [ Terrain decimated, all texture detail captured by RTT

# Results



# Results



# More

- [ Biggest problem: avoid blowing texture budget
- [ Had to arrange alpha testing and poly order to allow decent rendering
- [ Need to avoid walls separating from terrain: constrained Delauney triangulation
- [ Basic lesson: RTT is a valuable tool in game-generated LOD'ing.

# Spore: Plants

— [ First level is easy:

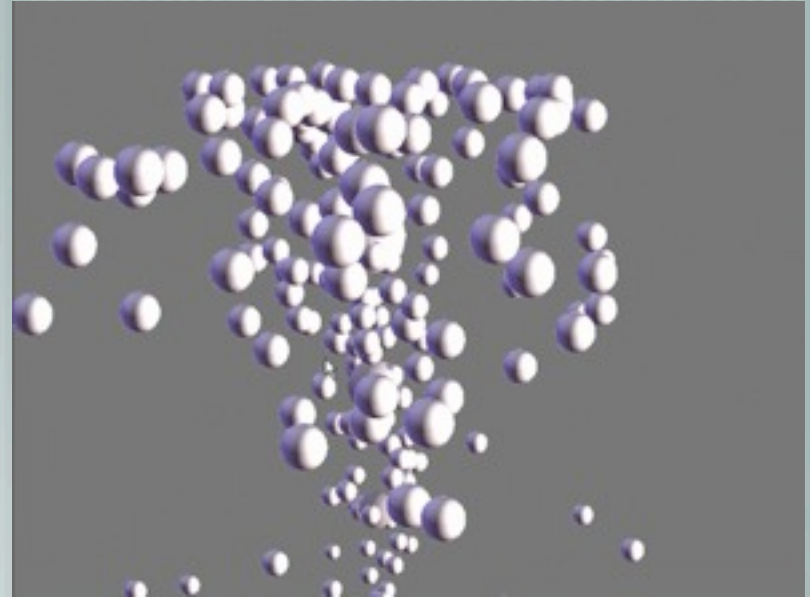
- Plants are combinations of branches (metaballs) and leaves (parts)
- Parts have pre-authored LOD
- Metaballs have natural LOD: tweak grid constants

— [ But what about a forest?

# Spore: Distant Plants

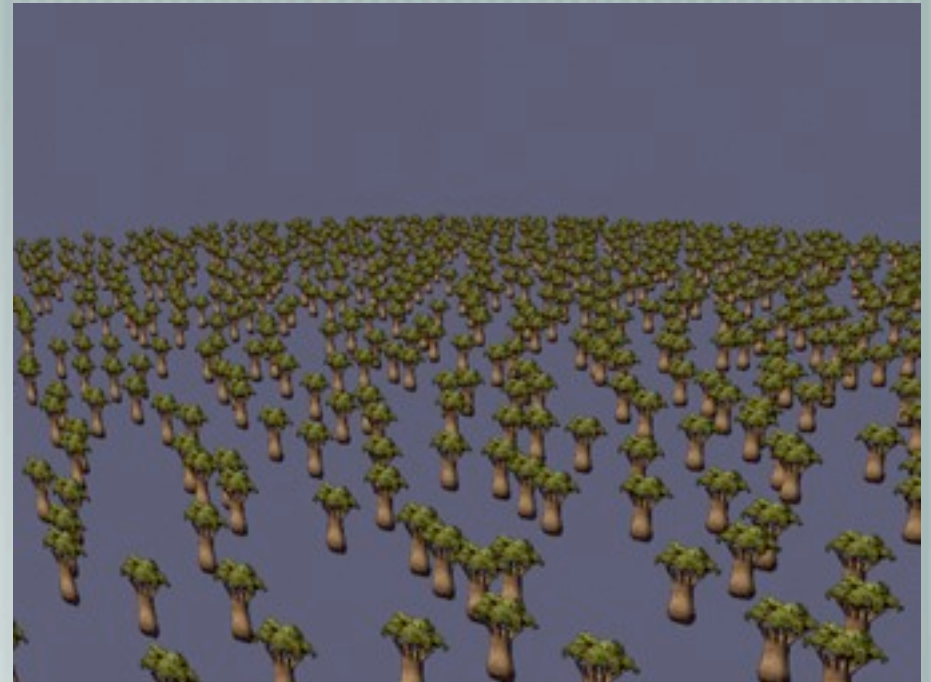
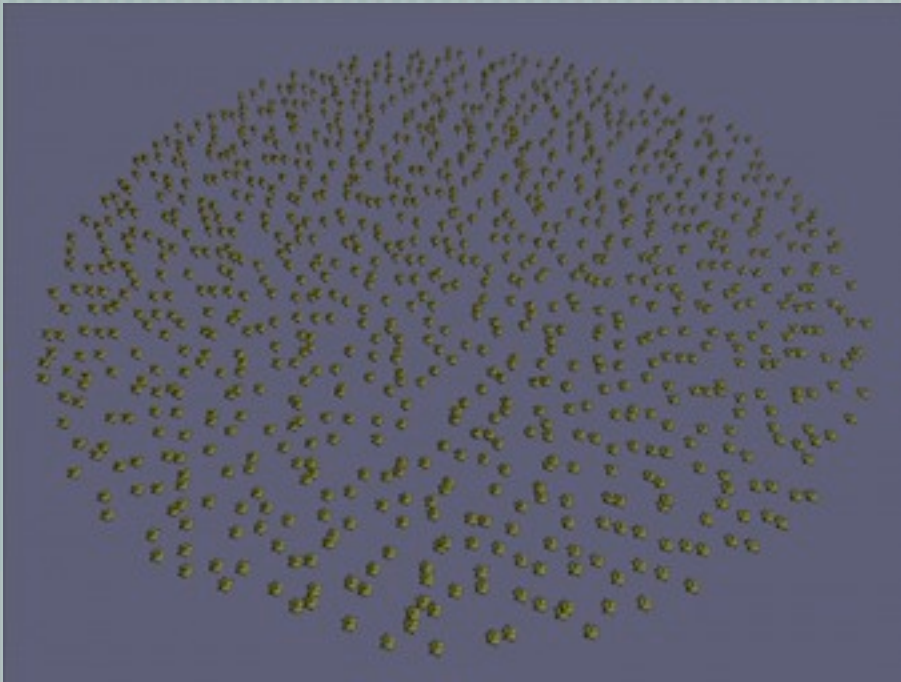
- [ Capture “cross tree” style imposter automatically
- [ Both diffuse texturing and normal map
- [ Normal-mapped particles/imposters work pretty well
  - Roll in ambient occlusion or bent normals if possible
- [ In general, lot to be gained from going to next level: capture more than just diffuse colour

# Imposters/NMap Particles

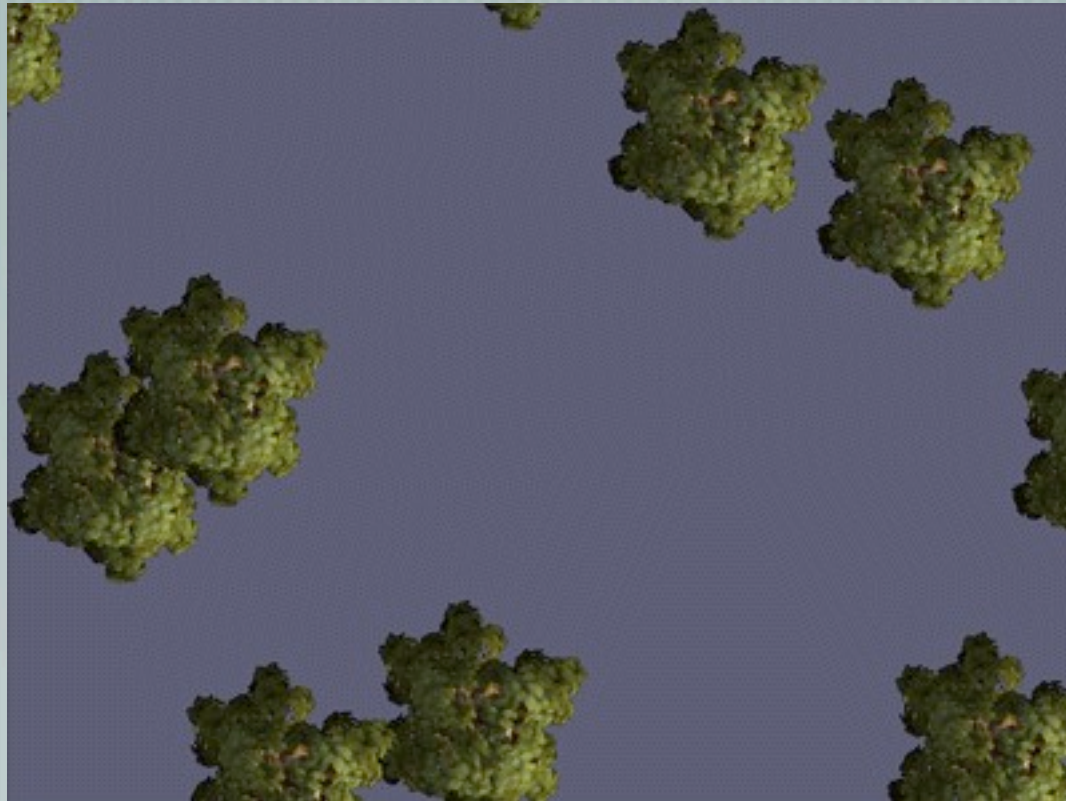


Two great tastes

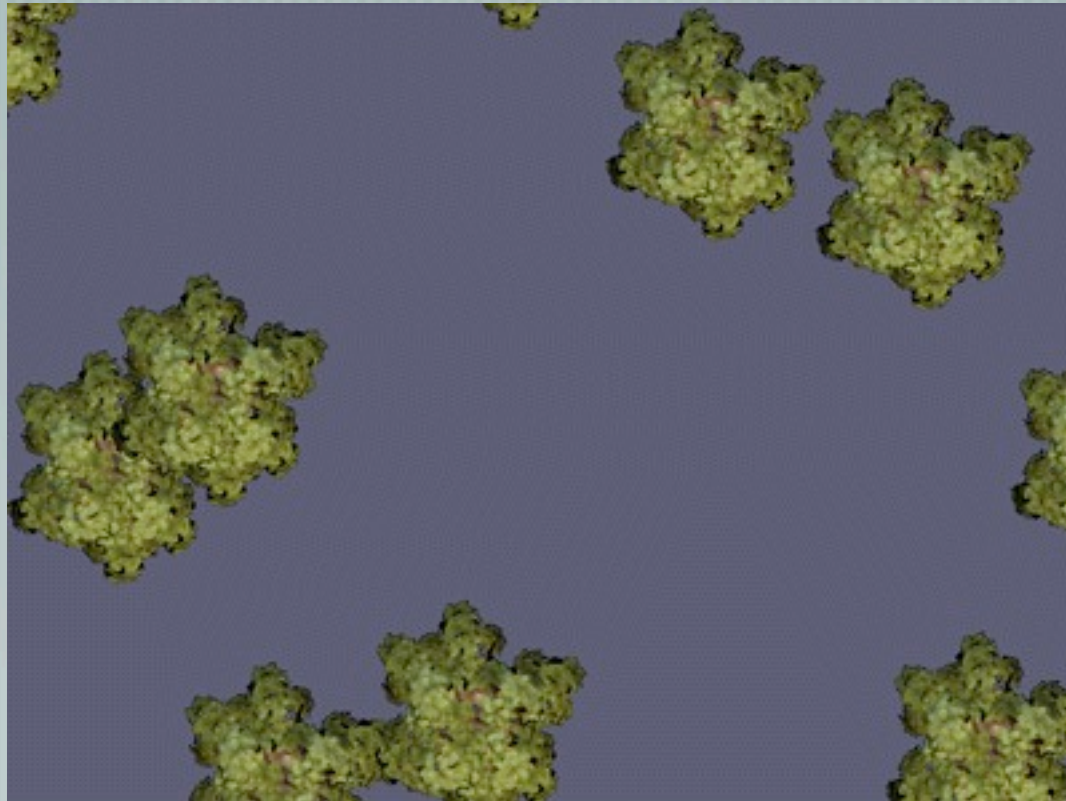
# Plant Imposters



# Plant Capture Example



# Plant Capture Example



# Now What?

- [ Bake instanced cross-trees, with same variations as applied to “real” objects, into single VB: fast render
- [ Cross fade, but must use alpha test
  - Set up vertex alpha so rendered quad shrinks off-axis
  - Easier: add noise to alpha channel (GPG2)
- [ So far so good, but could use Decoret et al.’s Billboard Clouds

# The Sims 2: Lighting

- [ User generates arbitrary amplified world – how do we light it?
- [ Can imagine facing the same problem in a completely procedurally generated world too.
- [ Have to get meta: your lighting system shifts up a level from direct lighting placement to a series of rules.
- [ Your lighter then tunes those rules to give good results.

# Rooms and Portals



Portals



Room Light Rig

# Rooms and Portals

- [ Lighting was room based

- Each room had a light rig generated for it automatically

- User-placed lights only affected objects in that room

- Portals transmitted light between rooms

- [ Time of day states, smooth transitions

# Tuning

- [ Exterior lighting predefined
- [ Objects, portals: various inside/outside light multipliers
  - Falloff, cutoff, intensity, directionality, etc.
  - 2x “over-bright” lighting
- [ Could overemphasize light directionality to create gradients on Sims, objects

# Light Mapping

- [ All floor/wall lighting lightmapped: 2D diffusion algorithm
- [ Don't be afraid of constructing light maps on the fly
  - Can thread in background, split up into workable chunks
  - Most delays covered by performing a smooth transition.  
Much less noticeable than pause-and-pop
- [ Cheap “irradiance” lighting: take gradient of light map

# Details

- [ Diffusion algorithm established nice gradients across rooms
- [ Room lighting rig: approximate light mapped environment
  - Ambient and small set of “delta” directional lights
  - Ambient sourced from light map for Sims

# Acknowledgements

— [ Ocean Quigley

— [ SC4: Christian Stratton, David Patch

— [ TS2: Don Livingston, Kevin Byall, Sam Yip, Christian Stratton,  
Jamie Kirschenbaum, Steve Kody

— [ Spore: Mike Khoury, Chris Hecker, Kent Jolly, Justin Graham